**B.M.S. COLLEGE OF ENGINEERING BENGALURU**
Autonomous Institute, Affiliated to VTU

Lab Record

# Software Engineering and Object-Oriented Modeling

*Submitted in partial fulfillment for the 6th Semester Laboratory*

Bachelor of Engineering
in
Computer Science and Engineering

*Submitted by:*

**CHINMAYI**

1BM21CS045

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Mar-June 2024

## B.M.S. COLLEGE OF ENGINEERING

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## *CERTIFICATE*

This is to certify that the Object-Oriented Analysis and Design(22CS6PCSEO) laboratory has been carried out by CHINMAYI (1BM21CS045) during the 6th Semester Mar-June-2024.

Signature of the Faculty Incharge:

SURABHI S

Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Table of Contents

# 1. Hotel Management System

1 Introduction:

1.1 Purpose of this Document: The purpose of this document is to outline the requirements and specifications for the development of a Hotel Management System. It will provide a clear understanding of the project objectives, scope, and deliverables.

1.2 Scope of this Document: This document defines the overall working and main objectives of the Hotel Management System. It includes a description of the development cost and time required for the project.

1.3 Overview: The Hotel Management System is a software solution designed to streamline hotel operations, including reservation management, guest check-in/check-out, room assignment, billing, and reporting.

2 General Description: The Hotel Management System will cater to the needs of hotel staff and management, providing features such as room booking, guest profiles, inventory management, and financial reporting. It will be accessible to users with varying levels of technical expertise.

3 Functional Requirements:

3.1 Reservation Management:

- Allow users to make room reservations online or through the front desk.
- Generate reservation confirmations and send notifications to guests.

3.2 Room Management:

- Assign rooms to guests based on availability and preferences.
- Track room status (clean, occupied, vacant) in real-time.

3.3 Guest Management:

- Maintain guest profiles with personal information, preferences, and booking history.
- Facilitate guest check-in and check-out processes.

### 3.4 Billing and Invoicing:

- Generate accurate bills for room charges, additional services, and taxes.
- Accept various payment methods and generate invoices for corporate clients.

## 4 Interface Requirements:

### 4.1 User Interface:

- Intuitive and user-friendly interface for hotel staff and guests.
- Accessible via web browsers, mobile devices, and desktop applications.

### 4.2 Integration Interfaces:

- Integration with payment gateways for secure transactions.
- Integration with third-party booking platforms for seamless reservation management.

## 5 Performance Requirements:

### 5.1 Response Time:

- The system should respond to user actions within 2 seconds.

### 5.2 Scalability:

- Handle a minimum of 1000 concurrent users during peak hours.

### 5.3 Data Integrity:

- Ensure data consistency and accuracy across all modules.

## 6 Design Constraints:

### 6.1 Hardware Limitations:

- The system should be compatible with standard hotel hardware (computers, printers, POS terminals).

### 6.2 Software Dependencies:

- Utilize a relational database management system (e.g., MySQL) for data storage.
- Use programming languages and frameworks conducive to UML modeling (e.g., Java, Spring Boot).

## 7 Non-Functional Attributes:

### 7.1 Security:

- Implement robust authentication and authorization mechanisms to protect sensitive data.

### 7.2 Reliability:

- Ensure high availability and fault tolerance to minimize system downtime.

### 7.3 Scalability:

- Design the system to accommodate future growth and expansion.

### 7.4 Portability:

- Support multiple platforms and devices for user accessibility.

### 7.5 Usability:

- The system shall have a user-friendly interface with clear navigation.

### 7.6 Reusability:

- The system shall use modular code design to facilitate future enhancements and maintenance.

### 7.7 Compatibility:

- The system shall be compatible with common web browsers (Chrome, Firefox, Safari

### 7.8 Data Integrity:

- The system shall ensure accurate and consistent data storage and retrieval.

8 Preliminary Schedule and Budget: The development of the Hotel Management System is estimated to take 6 months with a budget of $100,000. This includes project planning, development, testing, and deployment phases.

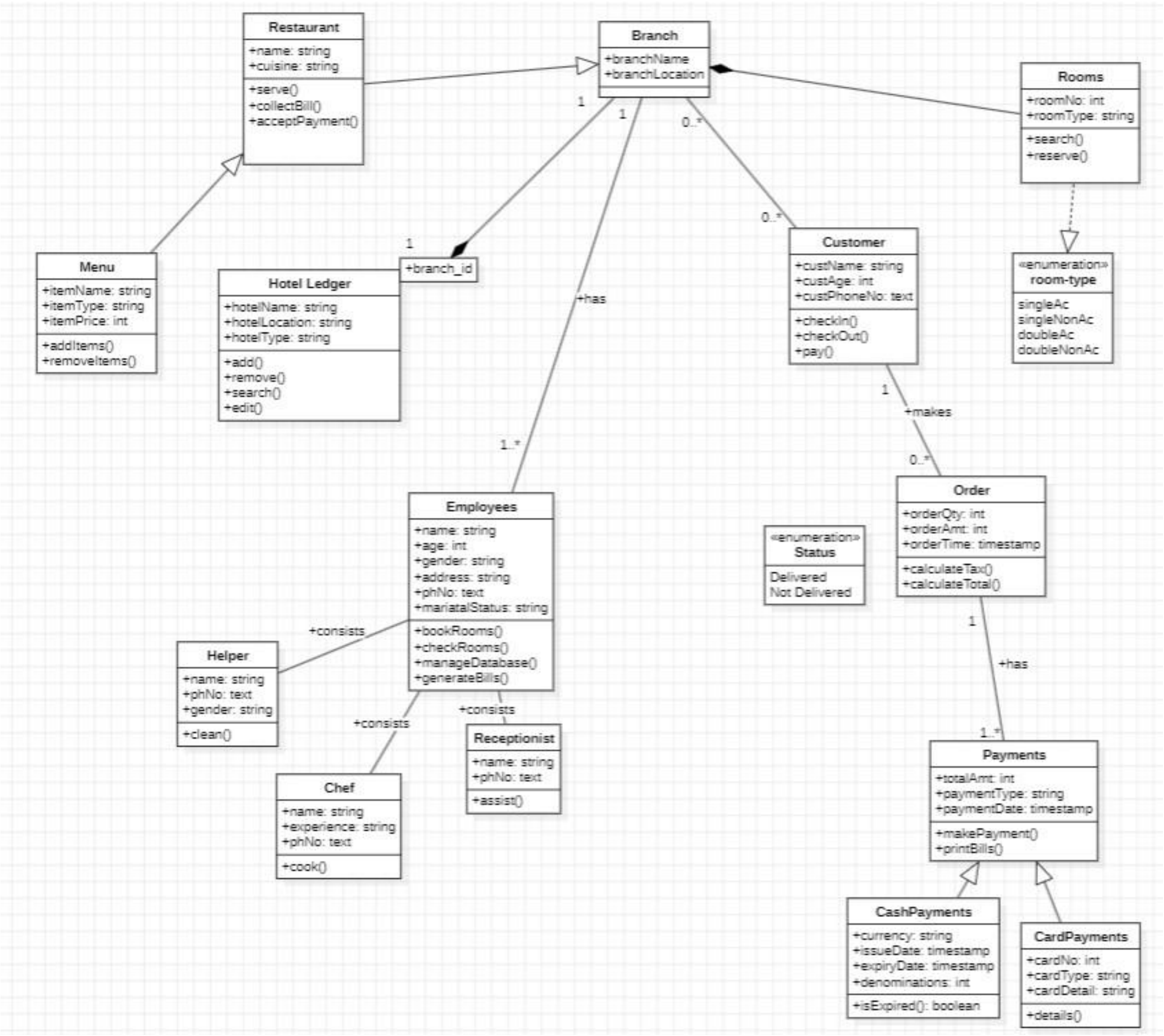

Fig 1.1: Hotel Management System Class Diagram

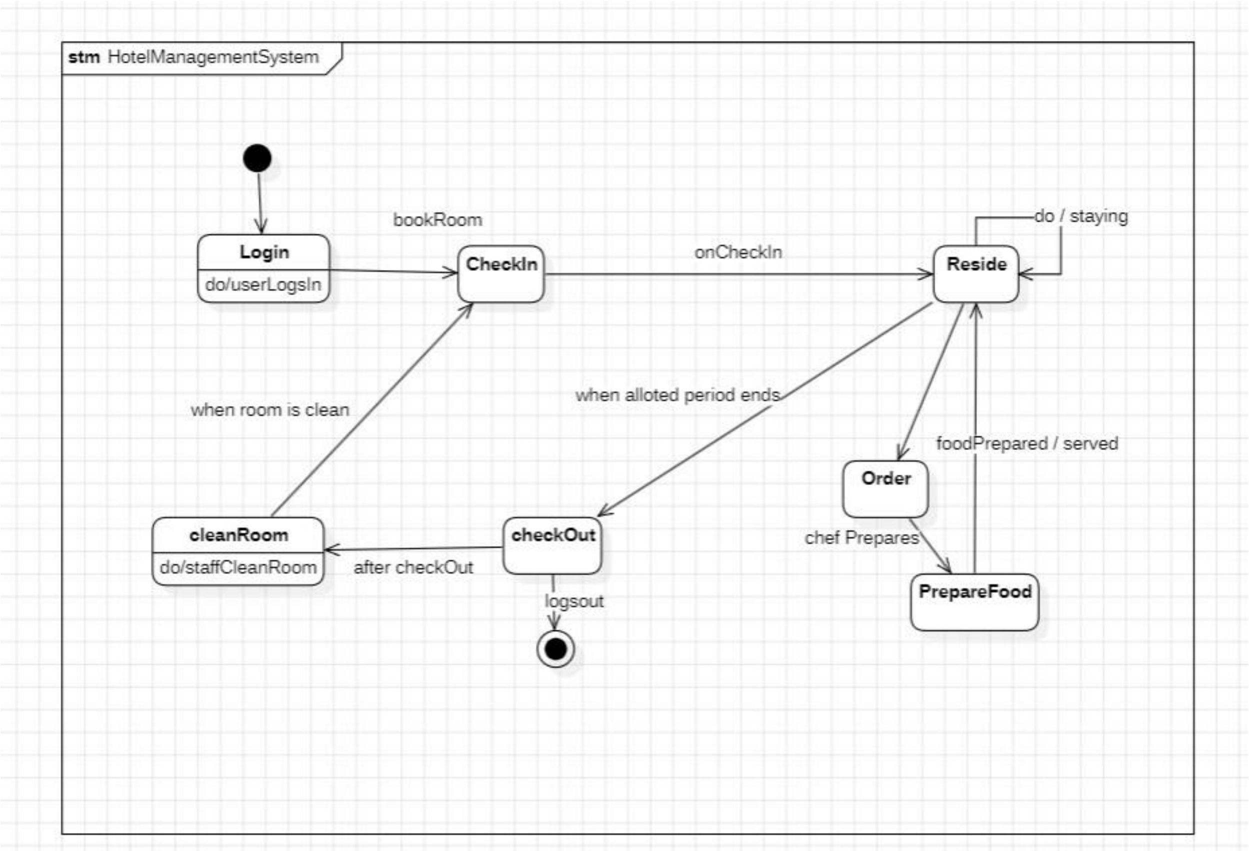

Fig 1.2: Hotel Management System State Diagram

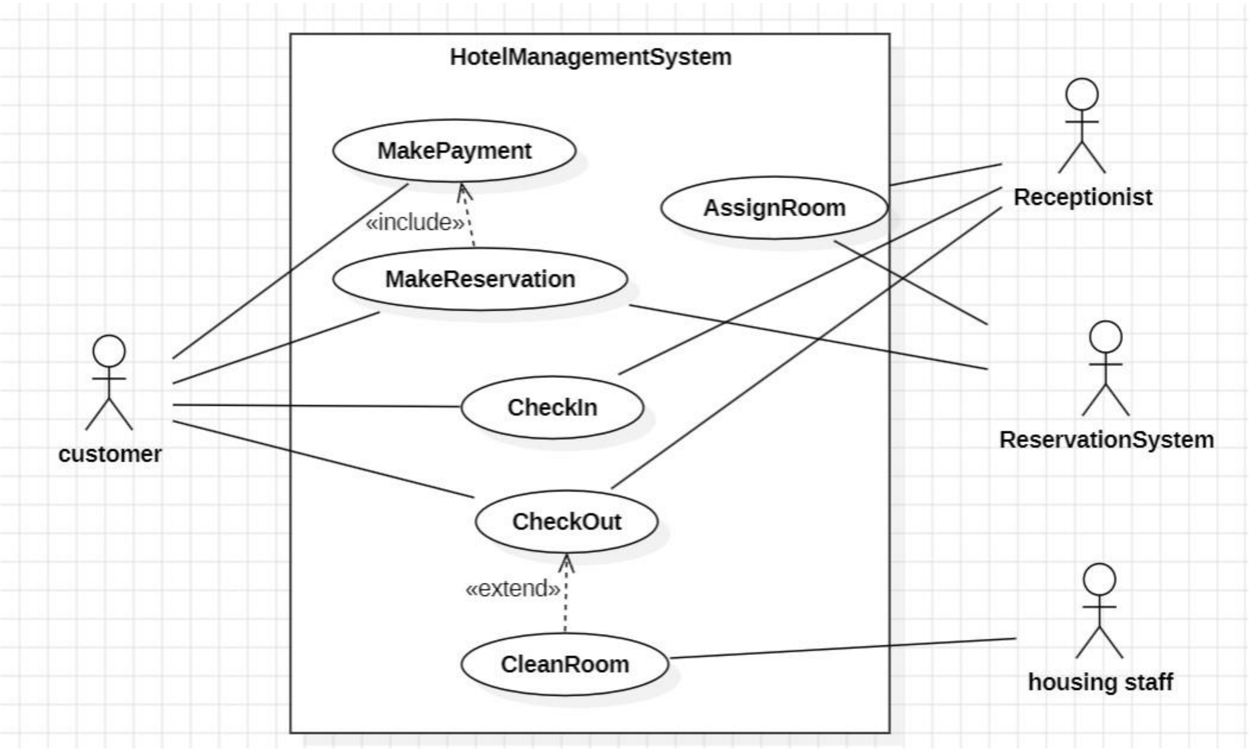

Fig 1.3: Hotel Management System Use Case Diagram

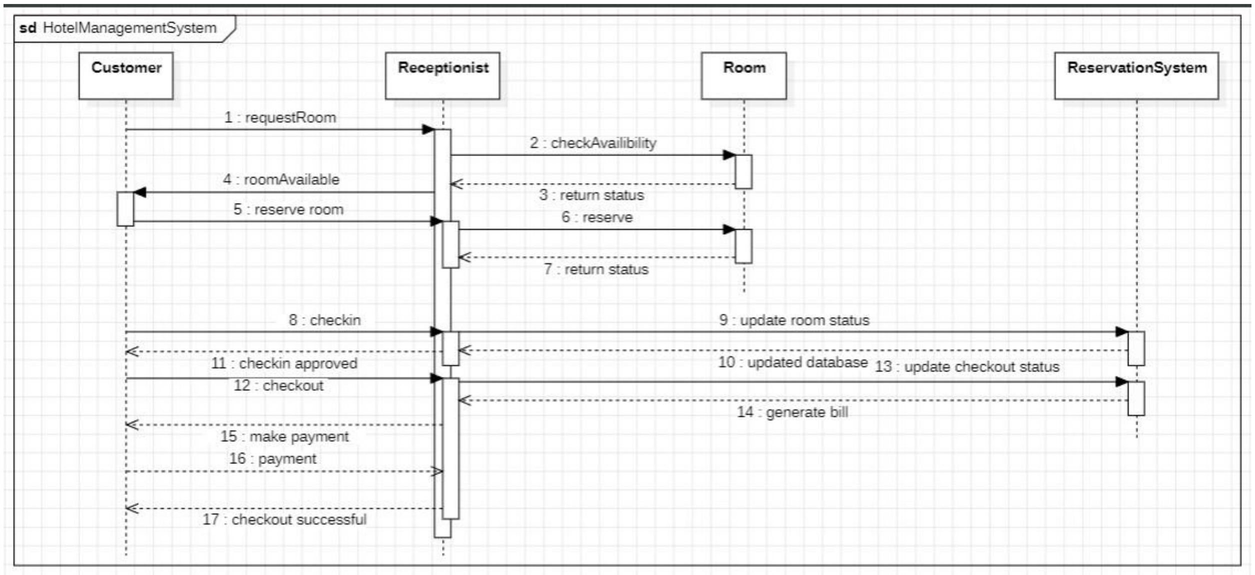

Fig 1.4: Hotel Management System Sequence Diagram

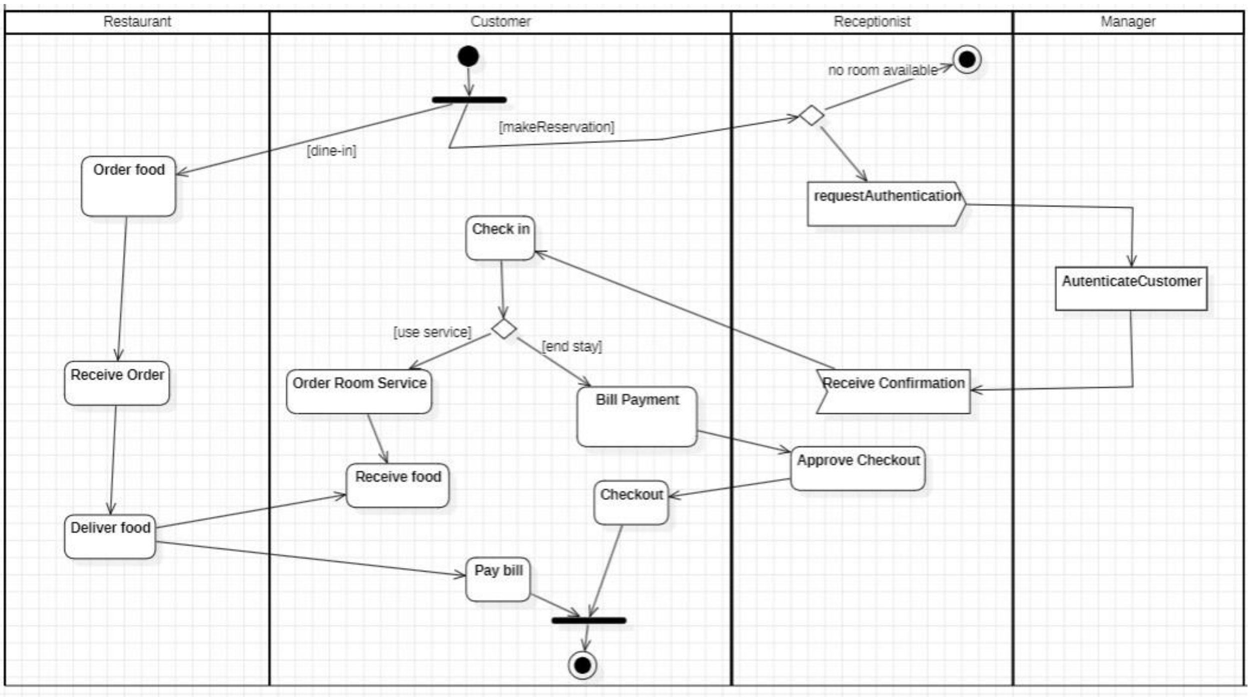

Fig 1.5: Hotel Management System Activity Diagram

# 2. Credit Card Processing

1 Introduction:

1.1 Purpose of this Document: The purpose of this document is to outline the requirements and specifications for the development of a Credit Card Processing System. It will provide a clear understanding of the project objectives, scope, and deliverables.

1.2 Scope of this Document: This document defines the overall working and main objectives of the Credit Card Processing System. It includes a description of the development cost and time required for the project.

1.3 Overview: The Credit Card Processing System is a software solution designed to facilitate secure and efficient processing of credit card transactions. It will handle authorization, capture, settlement, and reporting functionalities for merchants.

2 General Description:

The Credit Card Processing System will cater to the needs of merchants, providing features such as payment acceptance, transaction processing, fraud detection, and reporting. It will comply with industry standards and regulations for data security and privacy.

3 Functional Requirements:

3.1 Payment Acceptance:

- Accept credit card payments from customers through various channels (online, in-store, mobile).
- Support multiple payment methods (credit cards, debit cards, digital wallets).

3.2 Transaction Processing:

- Securely process transactions in real-time, ensuring accuracy and reliability.
- Validate cardholder information and perform authorization checks with card networks.

3.3 Fraud Detection:

- Implement fraud prevention measures such as address verification, CVV validation, and velocity checks.
- Monitor transactions for suspicious activity and trigger alerts for potential fraud.

3.4 Settlement and Reporting:

- Settle transactions with acquiring banks and payment processors in a timely manner.
- Generate detailed reports on transaction volume, revenue, chargebacks, and reconciliation.

4 Interface Requirements:

4.1 User Interface:

- Provide merchants with an intuitive dashboard for managing transactions, viewing reports, and accessing customer support.
- Offer APIs for seamless integration with existing point-of-sale systems and e-commerce platforms.

4.2 Integration Interfaces:

- Integrate with payment gateways, acquiring banks, and card networks for transaction processing.
- Support industry-standard protocols for secure communication and data exchange (e.g., SSL/TLS).

5 Performance Requirements:

5.1 Response Time:

- Process transactions within milliseconds to ensure a smooth checkout experience for customers.

5.2 Scalability:

- Handle a high volume of concurrent transactions during peak hours without performance degradation.

5.3 Reliability:

- Maintain high availability and uptime to prevent service disruptions and revenue loss.

6 Design Constraints:

6.1 Security Standards:

- Comply with PCI DSS requirements for storing, processing, and transmitting cardholder data securely.

6.2 Compliance:

- Adhere to regulatory requirements and industry standards (e.g., EMV, GDPR) for data protection and privacy.

7 Non-Functional Attributes:

7.1 Security:

- Implement robust authentication and authorization mechanisms to protect sensitive data.

7.2 Reliability:

- Ensure high availability and fault tolerance to minimize system downtime.

7.3 Scalability:

- Design the system to accommodate future growth and expansion.

7.4 Portability:

- Support multiple platforms and devices for user accessibility.

7.5 Usability:

- The system shall have a user-friendly interface with clear navigation.

7.6 Reusability:

- The system shall use modular code design to facilitate future enhancements and maintenance.

7.7 Compatibility:

- The system shall be compatible with common web browsers (Chrome, Firefox, Safari

7.8 Data Integrity:

- The system shall ensure accurate and consistent data storage and retrieval.

8 Preliminary Schedule and Budget: The development of the Credit Card Processing System is estimated to take 9 months with a budget of $500,000. This includes project planning, development, testing, certification, and deployment phases.

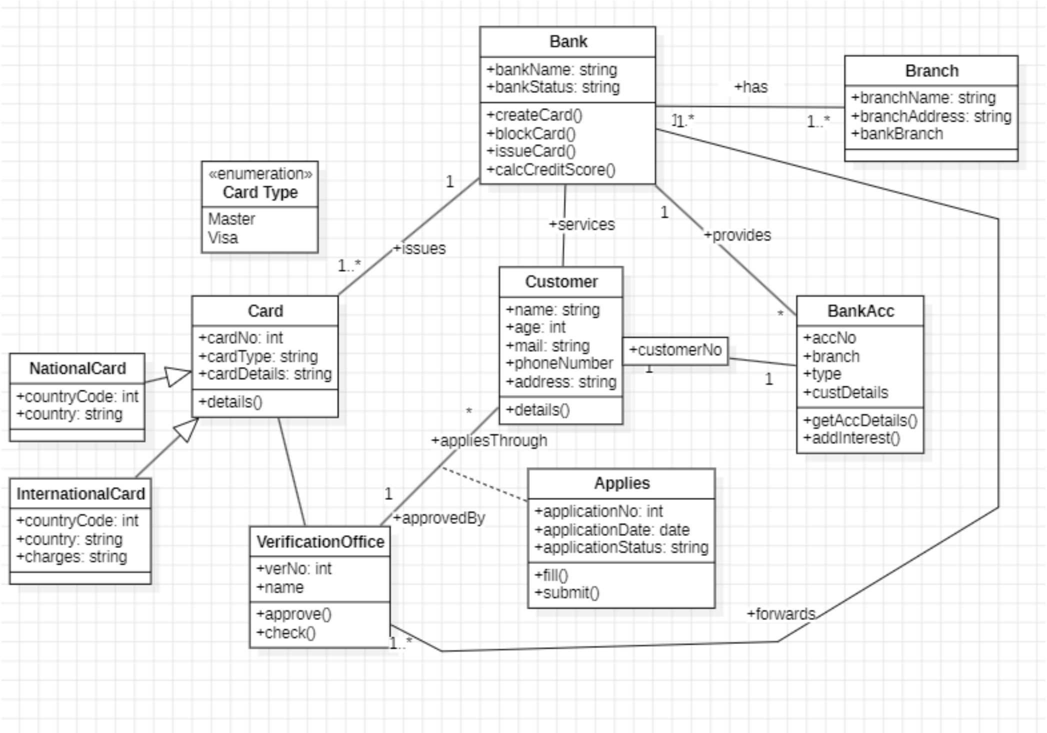

Fig 2.1: Credit Card Prcessing Class Diagram

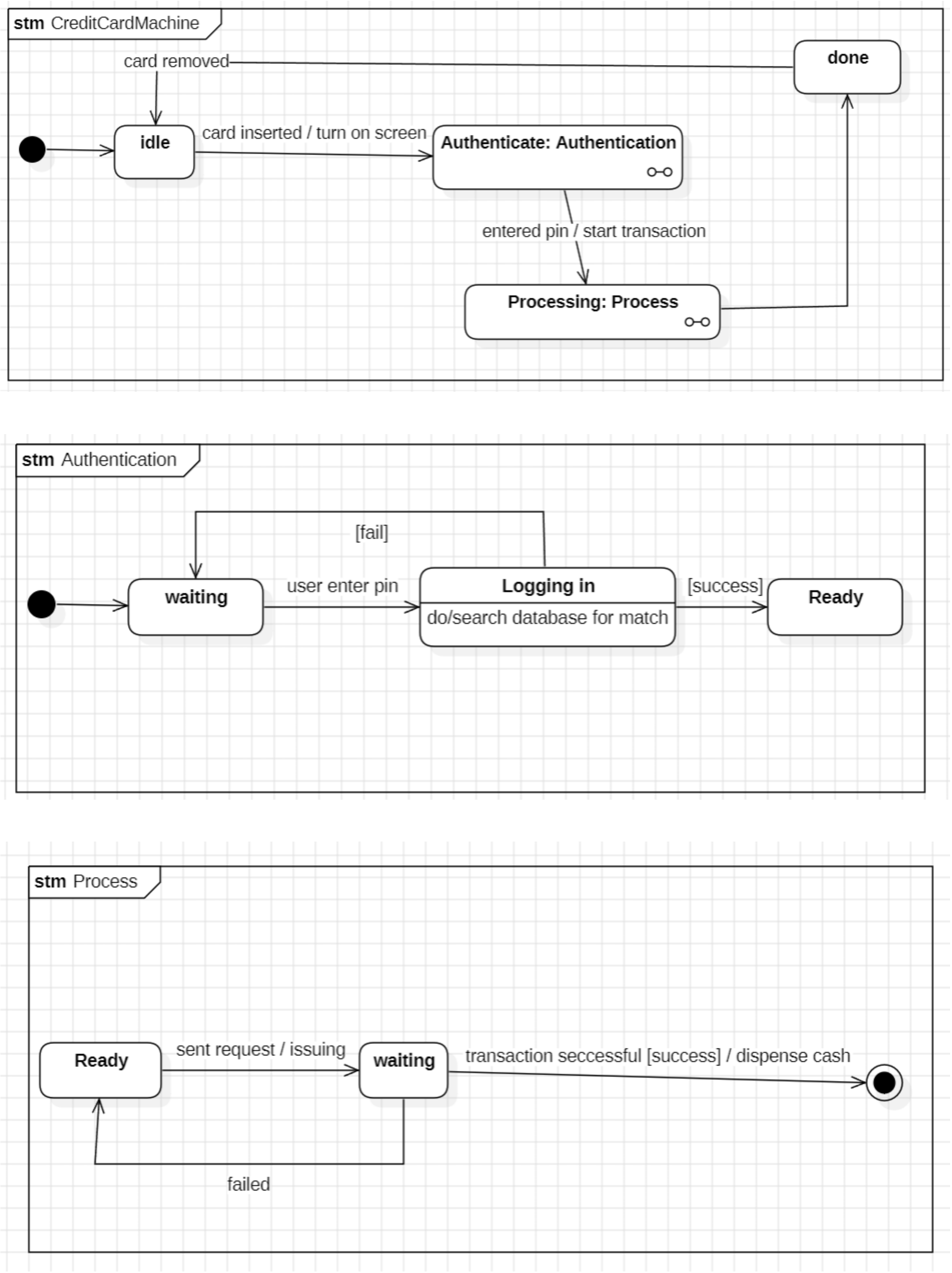

Fig 2.2: Credit Card Prcessing State Diagram

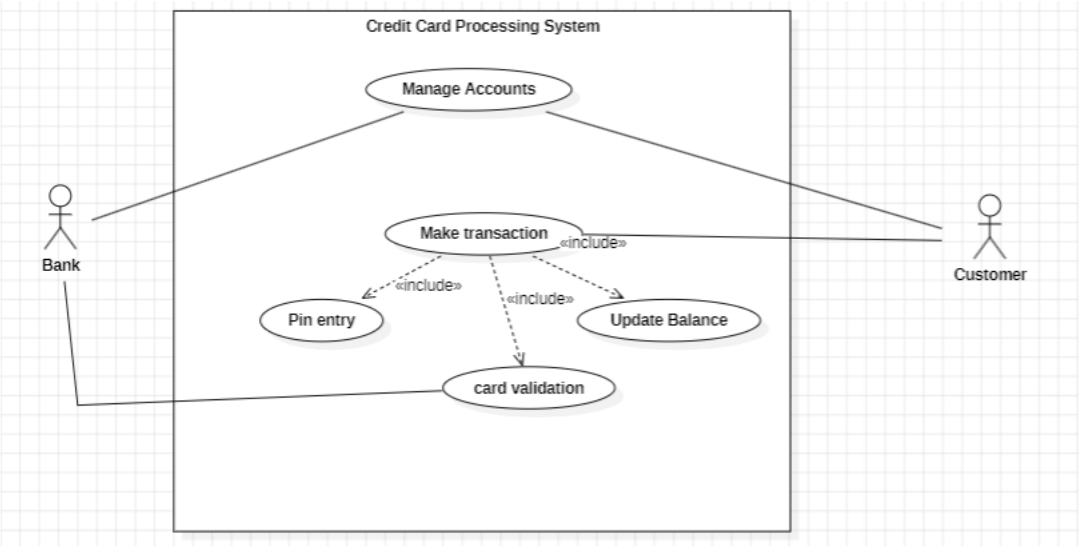

Fig 2.3: Credit Card Processing Use Case Diagram

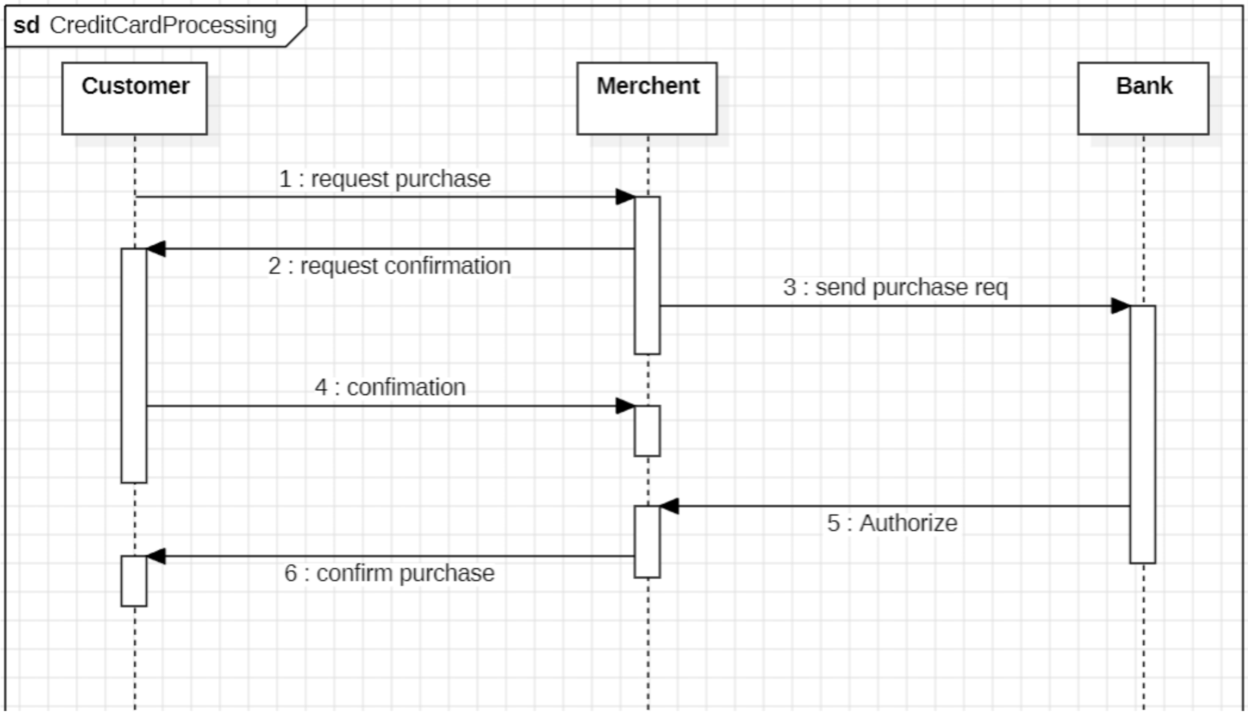

Fig 2.4: Credit Card Processing Sequence Diagram

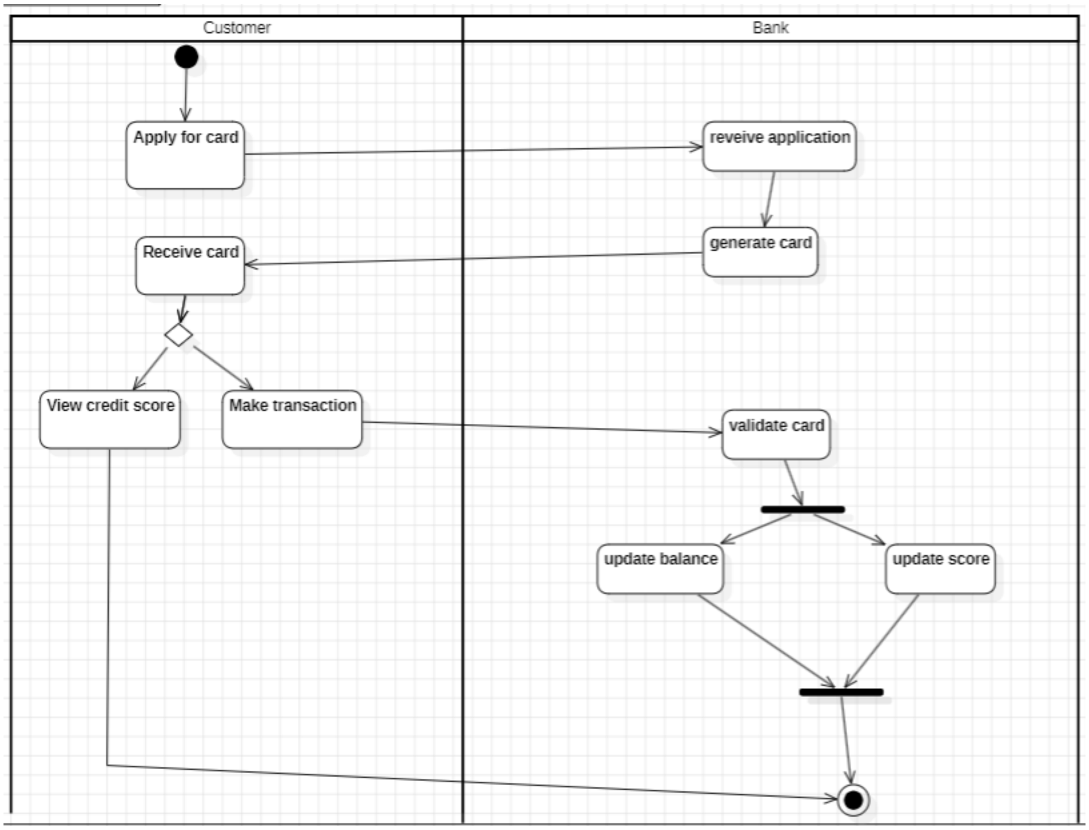

Fig 2.5: Credit Card Processing Activity Diagram

# 3. Library Management System

1 Introduction:

1.1 Purpose of this Document: The purpose of this document is to outline the requirements and specifications for the development of a Library Management System. It will provide a clear understanding of the project objectives, scope, and deliverables.

1.2 Scope of this Document: This document defines the overall working and main objectives of the Library Management System. It includes a description of the development cost and time required for the project.

1.3 Overview: The Library Management System is a software solution designed to automate library operations, including cataloging, circulation, patron management, and reporting. It aims to improve efficiency and user experience for librarians and patrons.

2 General Description:

The Library Management System will cater to the needs of librarians, library staff, and patrons, providing features such as book cataloging, borrowing, renewals, returns, reservations, and fine management. It will support both physical and digital collections.

The LMS will provide the following functionalities:

- Catalog Management: Catalog and manage books, journals, and other library materials.
- User Management: Handle user registrations, profiles, and authentication.
- Transaction Management: Manage borrowing, returning, and renewing of library materials.
- Fine Calculation: Calculate and manage overdue fines.
- Search Functionality: Allow users to search the catalog for available materials.
- Reporting: Generate reports on inventory, transactions, and user activity.

3 Functional Requirements:

3.1 Cataloging and Inventory Management:

- Allow librarians to add, edit, and delete book records with details such as title, author, ISBN, and genre.
- Track inventory levels, locations, and availability status of books.

3.2 Circulation Management:

- Enable patrons to check out and return books using self-service kiosks or librarian-assisted transactions.
- Implement borrowing rules, including loan periods, renewals, and holds/reservations.

3.3 Patron Management:

- Maintain patron profiles with personal information, contact details, and borrowing history.
- Enable librarians to issue library cards, update patron records, and manage fines.

3.4 Reporting and Analytics:

- Generate reports on library usage, circulation statistics, overdue items, and inventory status.
- Provide insights for collection development, budget planning, and resource allocation.

4 Interface Requirements:

4.1 User Interface:

- Provide intuitive interfaces for librarians to perform cataloging, circulation, and administrative tasks.
- Offer user-friendly portals for patrons to search for books, manage accounts, and access digital resources.

4.2 Integration Interfaces:

- Integrate with library databases, cataloging systems, and digital content providers for seamless data exchange.
- Support interoperability standards (e.g., MARC, SIP2) for integrating with third-party systems.

5 Performance Requirements:

5.1 Response Time:

- Ensure fast response times for searching, borrowing, and returning books to enhance user experience.

5.2 Scalability:

- Handle concurrent transactions and simultaneous access from multiple users without performance degradation.

5.3 Reliability:

- Maintain high availability and data integrity to prevent service disruptions and loss of library records.

6 Design Constraints:

6.1 Compatibility:

- Ensure compatibility with existing library infrastructure, including RFID systems, barcode scanners, and RFID tags.

6.2 Accessibility:

- Design the system to comply with accessibility standards (e.g., ADA) for users with disabilities.

7 Non-Functional Attributes:

7.1 Security:

- Implement robust authentication and authorization mechanisms to protect sensitive data.

7.2 Reliability:

- Ensure high availability and fault tolerance to minimize system downtime.

7.3 Scalability:

- Design the system to accommodate future growth and expansion.

7.4 Portability:

- Support multiple platforms and devices for user accessibility.

7.5 Usability:

- The system shall have a user-friendly interface with clear navigation.

7.6 Reusability:

- The system shall use modular code design to facilitate future enhancements and maintenance.

7.7 Compatibility:

- The system shall be compatible with common web browsers (Chrome, Firefox, Safari

7.8 Data Integrity:

- The system shall ensure accurate and consistent data storage and retrieval.

8 Preliminary Schedule and Budget: The development of the Library Management System is estimated to take 12 months with a budget of $750,000. This includes project planning, requirements analysis, design, development, testing, training, and deployment phases.

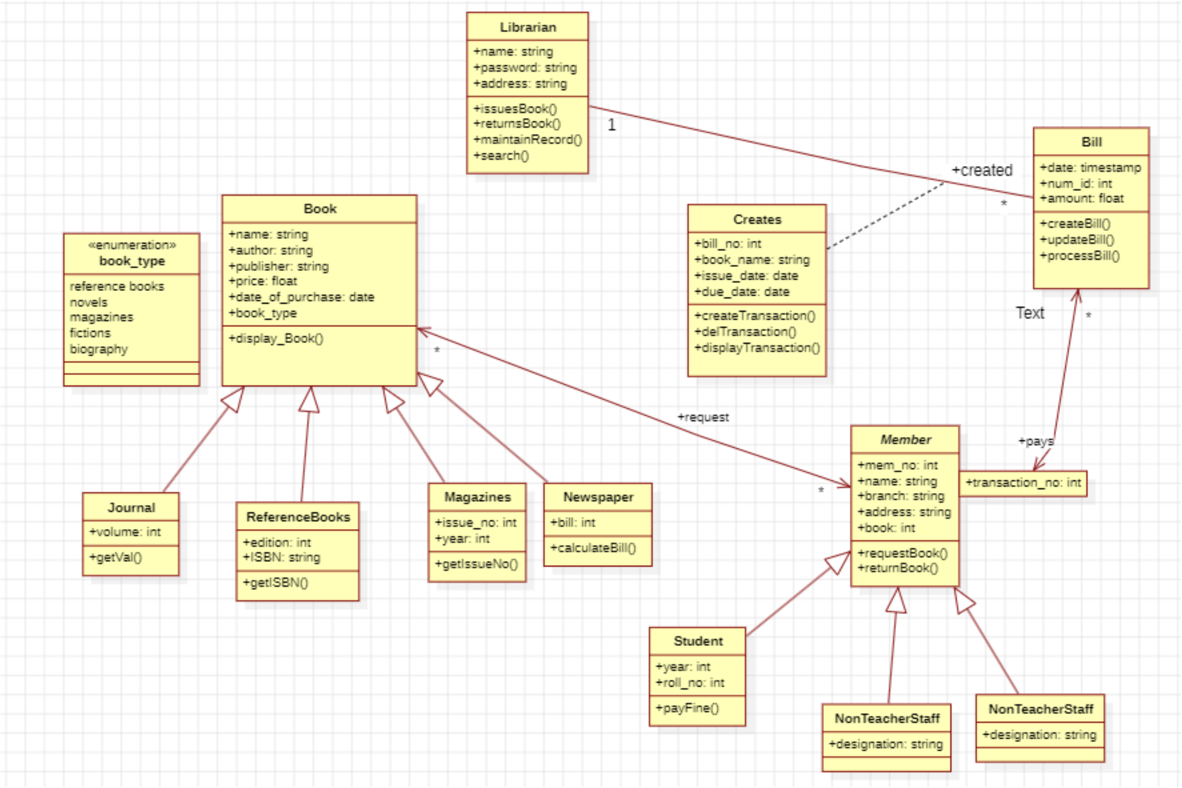

Fig 3.1: Library Management System Class Diagram

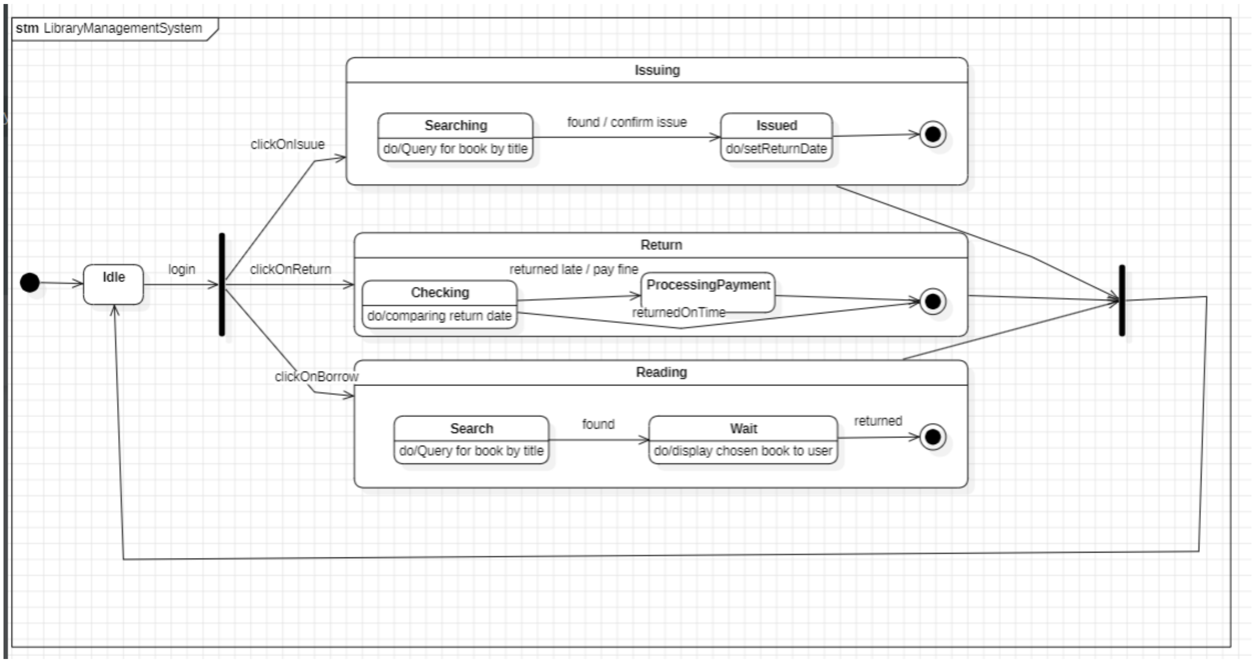

Fig 3.2: Library Management System State Diagram

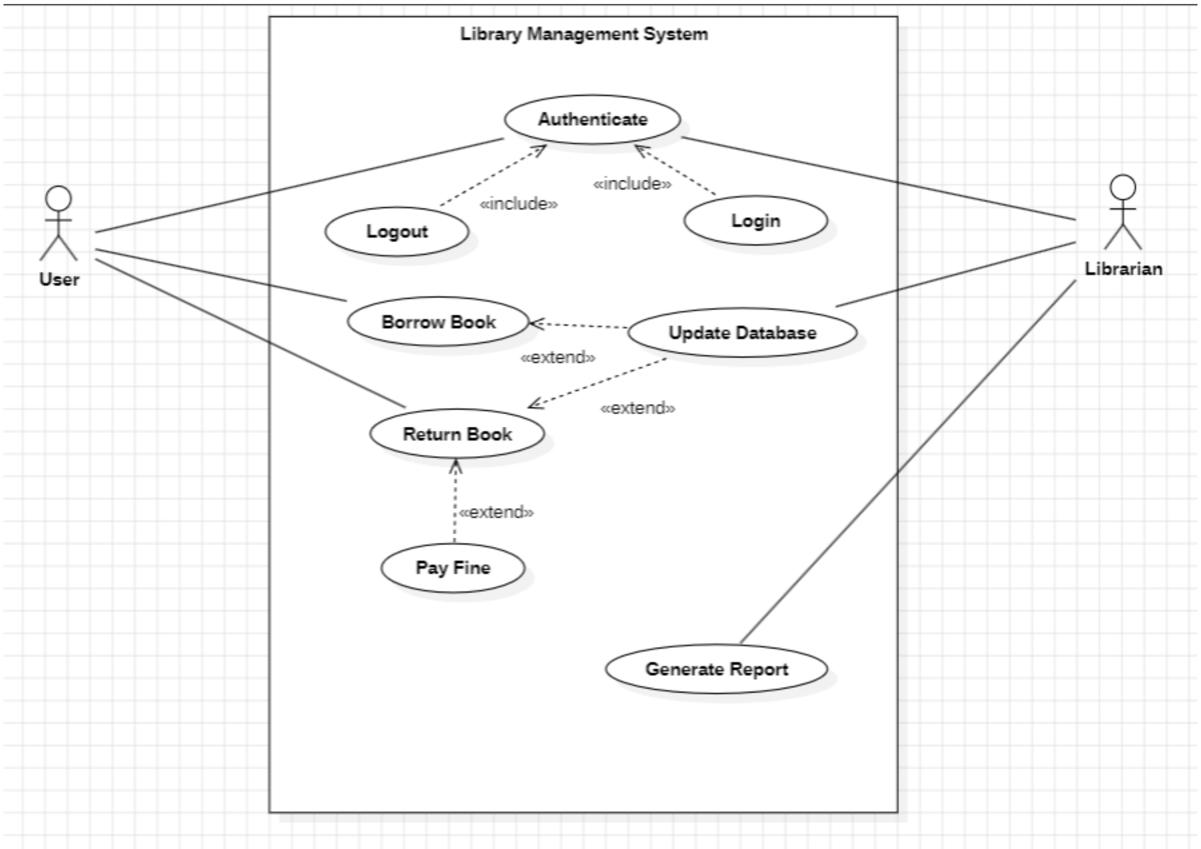

Fig 3.3: Library Management System Use Case Diagram

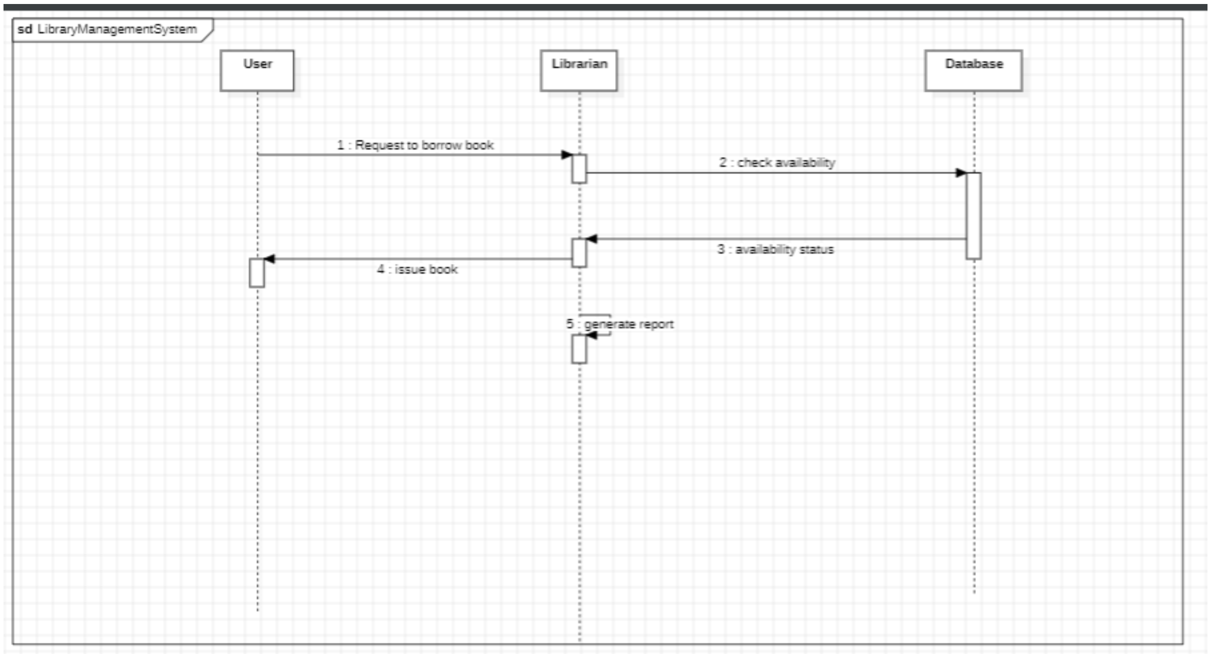

Fig 3.4: Library Management System Sequence Diagram

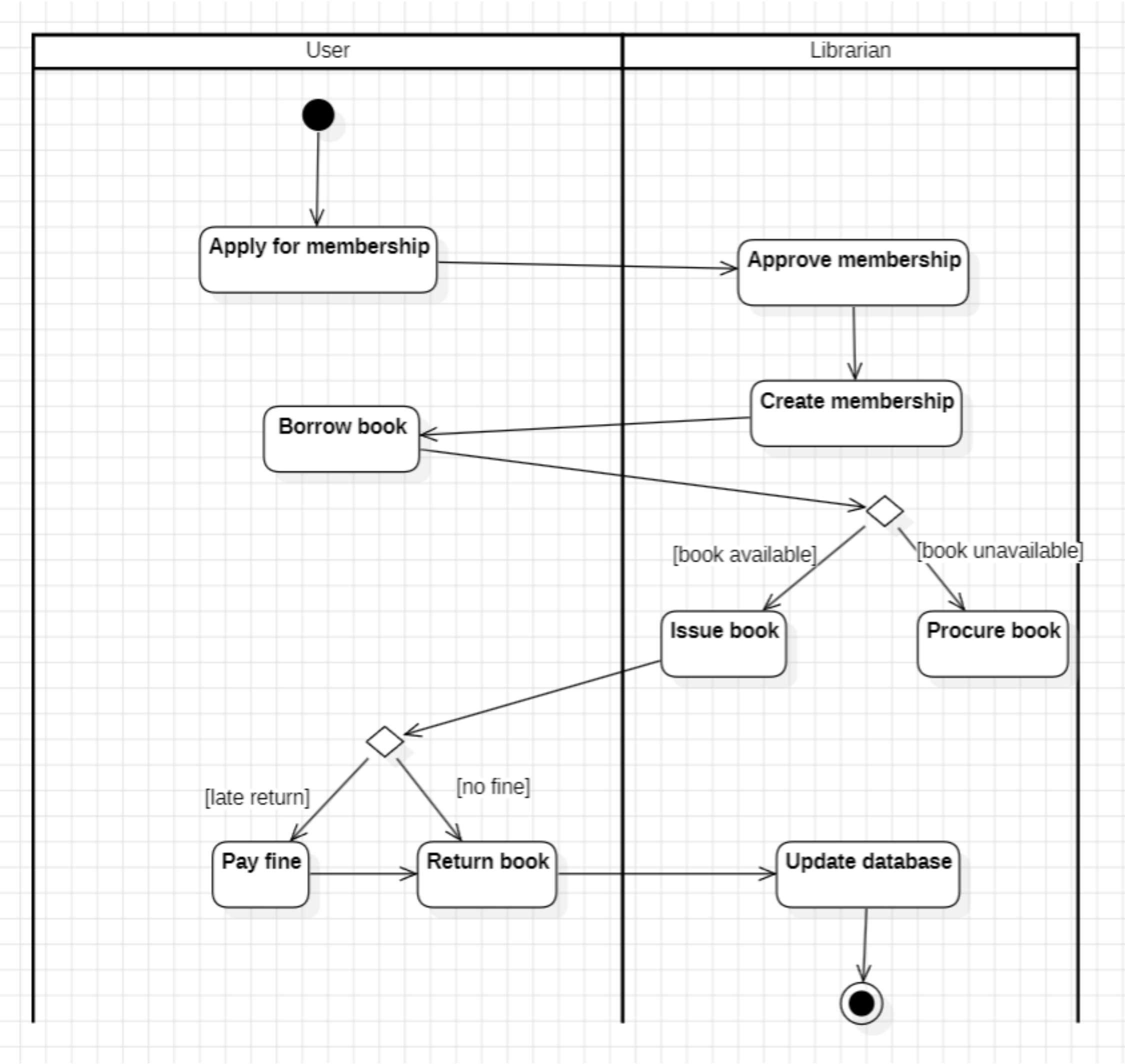

Fig 3.5: Library Management System Activity Diagram

# 4. Stock Maintenance System

1 Introduction:

1.1 Purpose of this Document: The purpose of this document is to outline the requirements and specifications for the development of a Stock Maintenance System. It will provide a clear understanding of the project objectives, scope, and deliverables.

1.2 Scope of this Document: This document defines the overall working and main objectives of the Stock Maintenance System. It includes a description of the development cost and time required for the project.

1.3 Overview: The Stock Maintenance System is a software solution designed to automate the management of inventory and stock levels for businesses. It will facilitate the tracking of stock items, inventory replenishment, stock transfers, and reporting.

2 General Description:

The Stock Maintenance System will cater to the needs of businesses, providing features such as stock item management, stock tracking, stock movement, and reporting. It will support various types of stock items, including raw materials, finished goods, and supplies.

3 Functional Requirements:

3.1 Stock Item Management:

- Allow users to add, edit, and delete stock items with details such as name, description, unit of measure, and cost.
- Maintain stock item categories and attributes for organization and classification.

3.2 Stock Tracking:

- Track stock levels in real-time, including quantities on hand, on order, allocated, and available.
- Provide visibility into stock movements, including receipts, issues, transfers, and adjustments.

3.3 Inventory Replenishment:

- Generate stock replenishment recommendations based on reorder points, lead times, and demand forecasts.
- Facilitate purchase order creation, submission, and approval for replenishing stock from suppliers.

3.4 Stock Movement:

- Support stock transfers between warehouses, locations, or departments within the organization.
- Record stock issues, returns, and adjustments to maintain accurate stock records.

4 Interface Requirements:

4.1 User Interface:

- Provide an intuitive interface for users to perform stock management tasks, view stock levels, and generate reports.
- Support role-based access control to restrict access to sensitive stock information based on user roles and permissions.

4.2 Integration Interfaces:

- Integrate with barcode scanners, RFID readers, or other data capture devices for efficient stock data entry and verification.
- Integrate with accounting systems, ERP systems, or third-party platforms for seamless data exchange and synchronization.

5 Performance Requirements:

5.1 Response Time:

- Ensure fast response times for stock inquiries, transactions, and reports to support real-time decision-making.

5.2 Scalability:

- Handle a growing number of stock items, transactions, and users without sacrificing system performance or reliability.

5.3 Reliability:

- Maintain high availability and data integrity to prevent stockouts, overstocks, and inventory discrepancies.

6 Design Constraints:

6.1 Compatibility:

- Ensure compatibility with existing hardware, software, and infrastructure components used for stock management and data capture.

6.2 Data Security:

- Implement encryption, access controls, and audit trails to protect sensitive stock information from unauthorized access, manipulation, or disclosure.

7 Non-Functional Attributes:

7.1 Security:

- Implement robust authentication and authorization mechanisms to protect sensitive data.

7.2 Reliability:

- Ensure high availability and fault tolerance to minimize system downtime.

7.3 Scalability:

- Design the system to accommodate future growth and expansion.

7.4 Portability:

- Support multiple platforms and devices for user accessibility.

7.5 Usability:

- The system shall have a user-friendly interface with clear navigation.

7.6 Reusability:

- The system shall use modular code design to facilitate future enhancements and maintenance.

7.7 Compatibility:

- The system shall be compatible with common web browsers (Chrome, Firefox, Safari

7.8 Data Integrity:

- The system shall ensure accurate and consistent data storage and retrieval.

8 Preliminary Schedule and Budget: The development of the Stock Maintenance System is estimated to take 9 months with a budget of Rs.500,000. This includes project planning, requirements analysis, design, development, testing, training, and deployment phases.

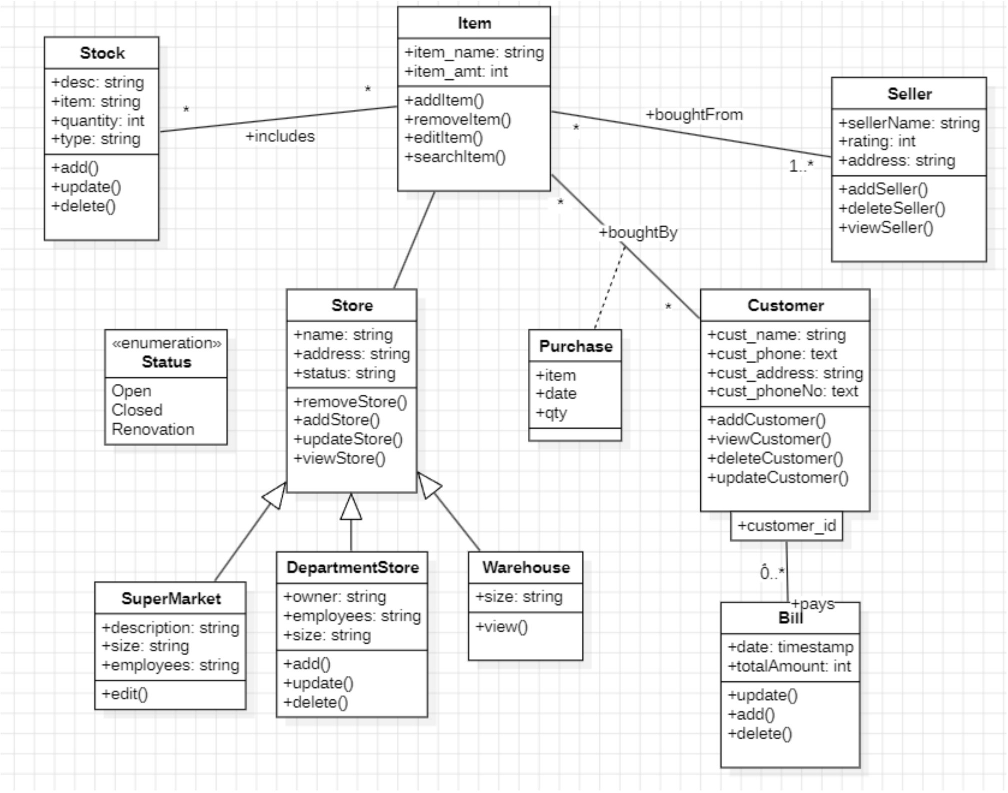

Fig 4.1: Stock Maintenance System Class Diagram

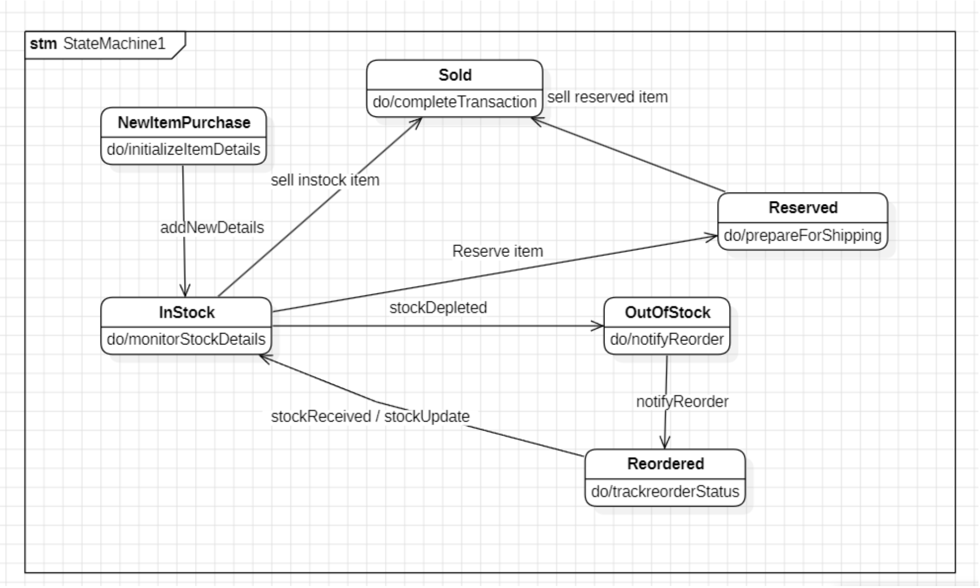

Fig 4.2: Stock Maintenance System State Diagram

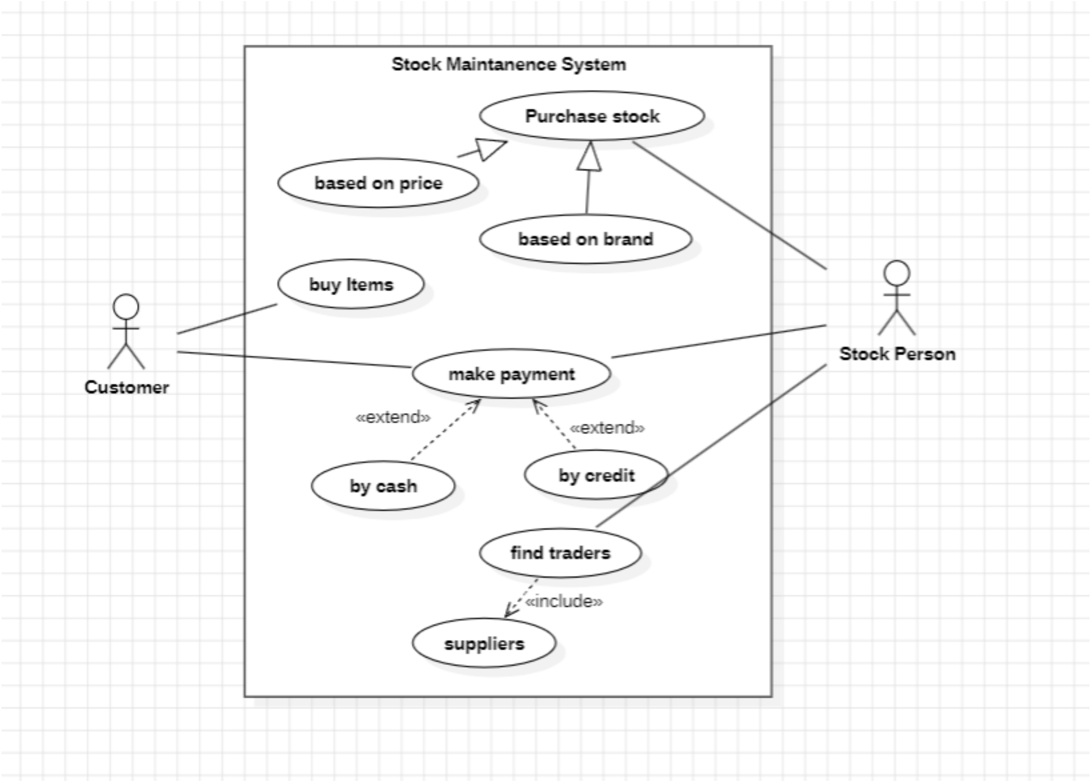

Fig 4.3: Stock Maintenance System Use Case Diagram

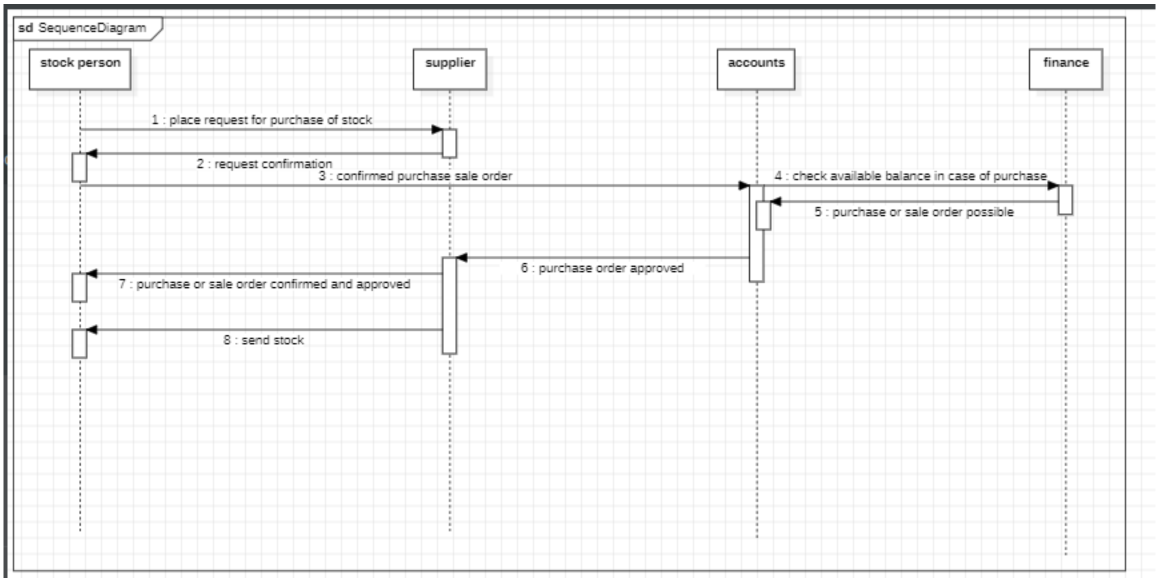
sd SequenceDiagram
stock person
supplier
accounts
finance
1 : place request for purchase of stock
2 : request confirmation
3 : confirmed purchase sale order
4 : check available balance in case of purchase
5 : purchase or sale order possible
6 : purchase order approved
7 : purchase or sale order confirmed and approved
8 : send stock

Fig 4.4: Stock Maintenance System Sequence Diagram

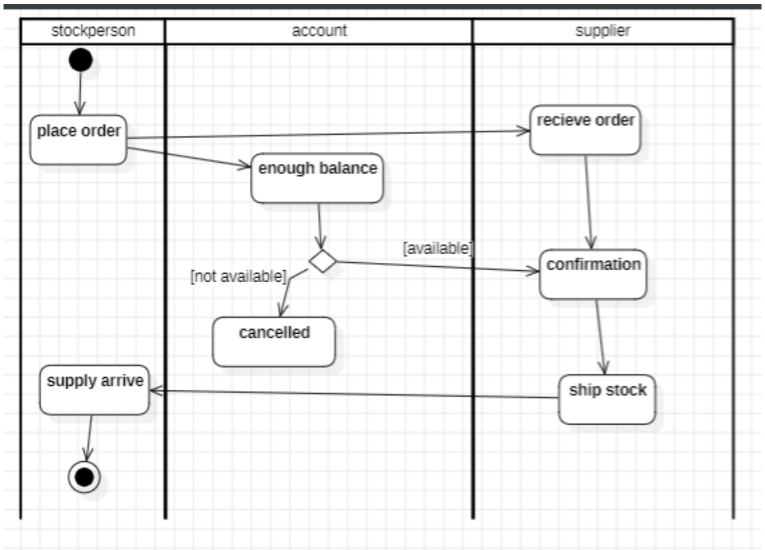

Fig 4.5: Stock Maintenance System Activity Diagram

# 5. Passport Automation System

1 Introduction:

1.1 Purpose of this Document: The purpose of this document is to outline the requirements and specifications for the development of a Passport Automation System. It will provide a clear understanding of the project objectives, scope, and deliverables.

1.2 Scope of this Document: This document defines the overall working and main objectives of the Passport Automation System. It includes a description of the development cost and time required for the project.

1.3 Overview: The Passport Automation System is a software solution designed to streamline the process of passport application, issuance, renewal, and tracking. It aims to enhance efficiency, accuracy, and security in managing passport-related services.

2 General Description:

The Passport Automation System will cater to the needs of passport authorities, applicants, and government agencies, providing features such as online application submission, appointment scheduling, document verification, payment processing, and passport delivery tracking.

3 Functional Requirements:

3.1 Online Application Submission:

- Enable applicants to submit passport applications online through a user-friendly interface, capturing personal details, biographic information, and supporting documents.
- Validate application data in real-time to ensure completeness and accuracy.

3.2 Appointment Scheduling:

- Allow applicants to schedule appointments for passport application processing, biometric data collection, and document verification at designated passport offices or centers.
- Manage appointment slots, availability, and capacity to optimize resource utilization.

3.3 Document Verification:

- Facilitate the verification of applicant documents, including identity proof, address proof, and supporting certificates, by passport officials or designated authorities.
- Record verification outcomes and update application status accordingly.

3.4 Payment Processing:

- Accept online payments for passport application fees, processing charges, and additional services through secure payment gateways or government payment portals.
- Generate payment receipts and invoices for applicant records and financial reconciliation.

3.5 Passport Issuance and Delivery:

- Generate passport documents and personalize them with applicant details, biometric data, and security features, following predefined issuance guidelines and standards.
- Track passport delivery status and provide applicants with real-time updates on dispatch, transit, and delivery milestones.

4 Interface Requirements:

4.1 User Interface:

- Provide intuitive interfaces for applicants to complete passport applications, schedule appointments, and track application status.
- Design user-friendly portals for passport authorities to manage application processing, document verification, and passport issuance tasks.

4.2 Integration Interfaces:

- Integrate with government databases, identity verification systems, and biometric authentication platforms to validate applicant information and ensure data integrity.
- Support interoperability with external systems for document scanning, payment processing, courier services, and postal delivery.

5 Performance Requirements:

5.1 Response Time:

- Ensure fast response times for online application submission, appointment scheduling, and document verification processes to minimize applicant wait times and enhance user experience.

5.2 Scalability:

- Handle a large volume of passport applications, appointment bookings, and document verifications concurrently, especially during peak periods or promotional campaigns.

5.3 Reliability:

- Maintain high availability and system uptime to prevent service disruptions and ensure continuous access to passport services for applicants and passport authorities.

6 Design Constraints:

6.1 Security Standards:

- Implement robust authentication, encryption, and access controls to protect sensitive applicant data, biometric information, and financial transactions from unauthorized access or misuse.

6.2 Regulatory Compliance:

- Ensure compliance with government regulations, data protection laws, and international standards (e.g., ICAO) governing passport issuance, biometric data handling, and cross-border travel security.

7 Non-Functional Attributes:

7.1 Security:

- Implement robust authentication and authorization mechanisms to protect sensitive data.

7.2 Reliability:

- Ensure high availability and fault tolerance to minimize system downtime.

7.3 Scalability:

- Design the system to accommodate future growth and expansion.

7.4 Portability:

- Support multiple platforms and devices for user accessibility.

7.5 Usability:

- The system shall have a user-friendly interface with clear navigation.

7.6 Reusability:

- The system shall use modular code design to facilitate future enhancements and maintenance.

7.7 Compatibility:

- The system shall be compatible with common web browsers (Chrome, Firefox, Safari

7.8 Data Integrity:

- The system shall ensure accurate and consistent data storage and retrieval.

8 Preliminary Schedule and Budget: The development of the Passport Automation System is estimated to take 12 months with a budget of $1,000,000. This includes project planning, requirements analysis, design, development, testing, training, deployment, and maintenance phases.

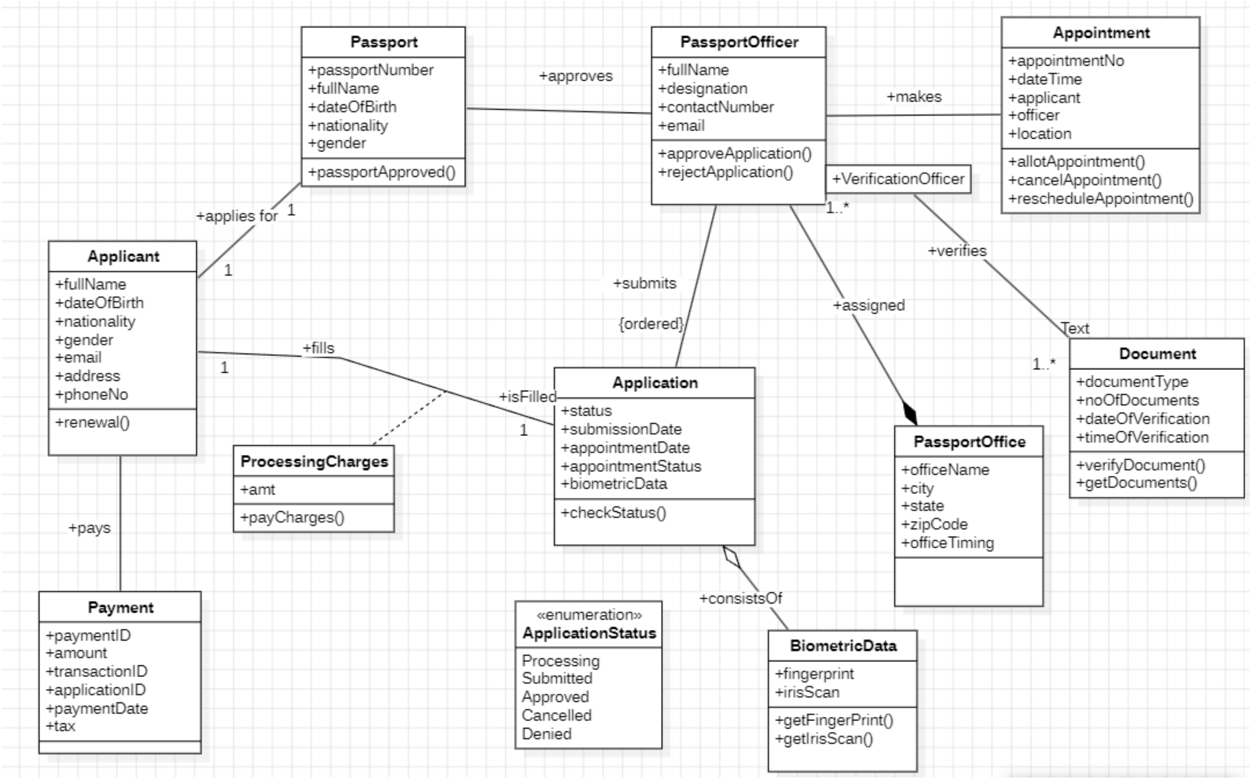

Fig 5.1: Passport Automation System Class Diagram

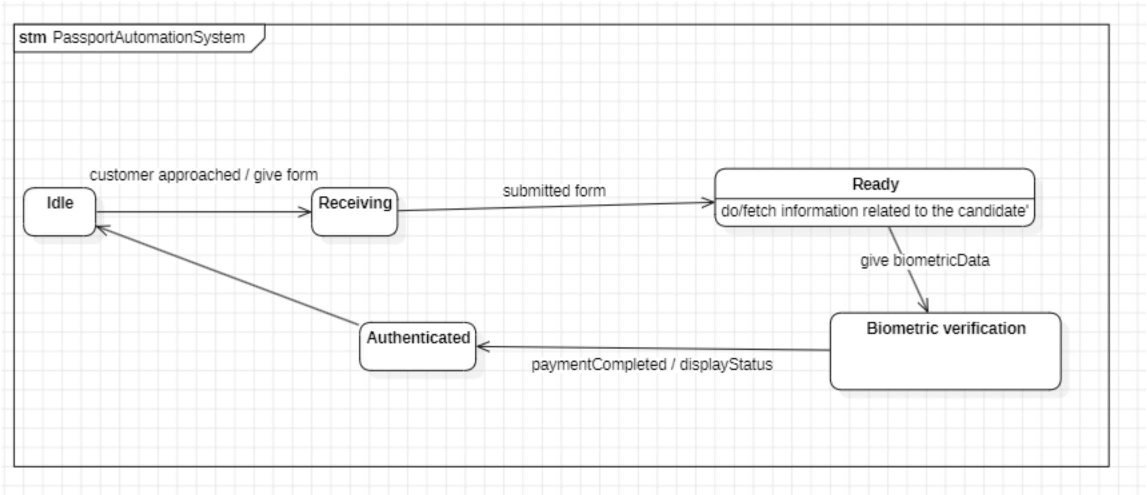
stm PassportAutomationSystem
customer approached / give form
Idle
Receiving
submitted form
Ready
do/fetch information related to the candidate'
give biometricData
Biometric verification
Authenticated
paymentCompleted / displayStatus

Fig 5.2: Passport Automation System State Diagram

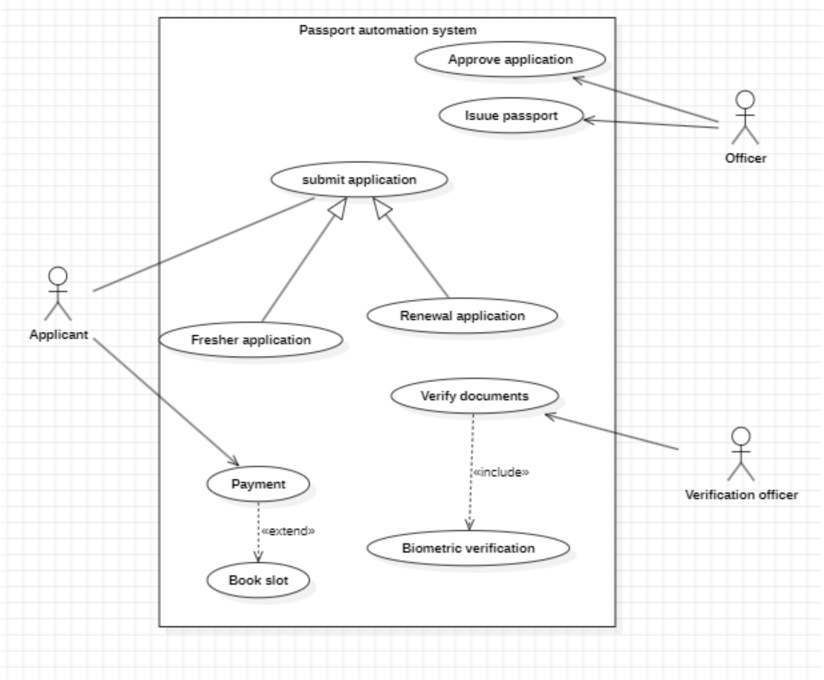

Fig 5.3: Passport Automation System Use Case Diagram

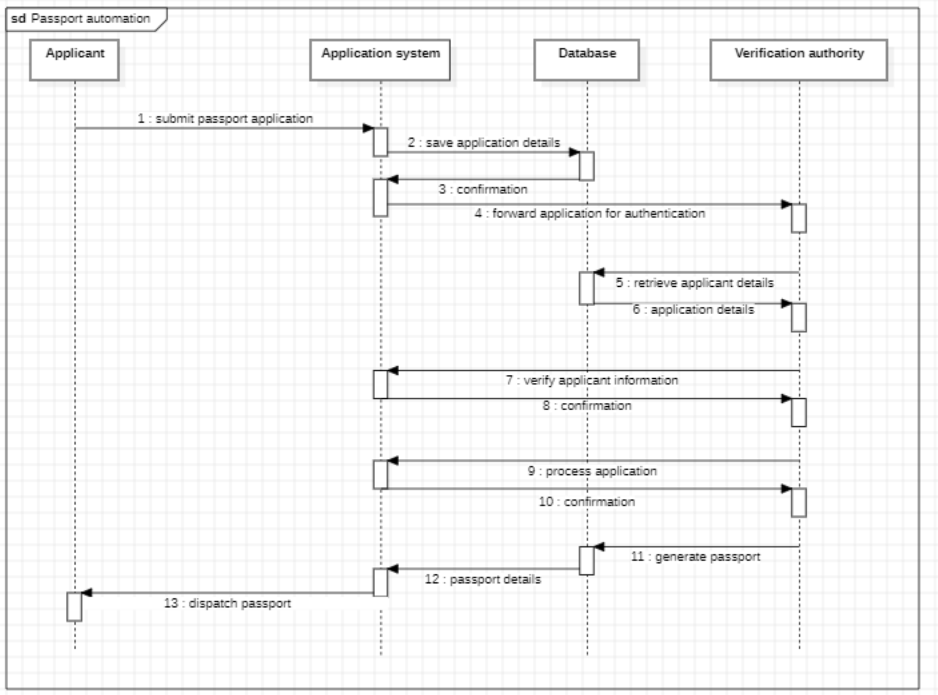

Fig 5.4: Passport Automation Sequence Diagram

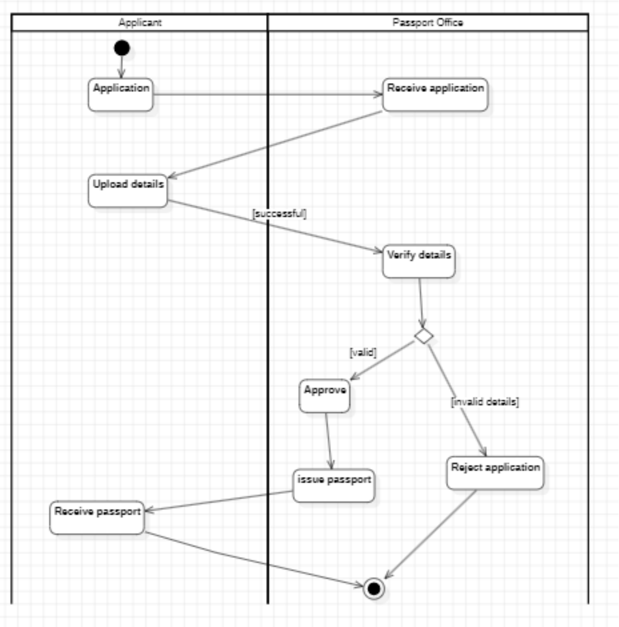

Fig 5.5: Passport Automation System Activity Diagram